

IN THE CLAIMS:

THIS LISTING OF CLAIMS WILL REPLACE ALL PRIOR VERSIONS, AND LISTINGS, OF CLAIMS IN THE APPLICATION. All claims are original except for claims 5 and 44 which are amended to correct for only minor antecedent reference misuse or typographical error.

LISTING OF CLAIMS:

1. (ORIGINAL) In a computer network having an events notification system, said network including at least one client having a client-database, at least one server having at least one server-database, said at least one client: (1) selecting said events and desired notification responses thereto to obtain selected data, (2) creating a plurality of event-notification template objects from said selected data and storing said objects in said client database; and, (3) transmitting said objects from said client database to selected ones of said at least one server to obtain server-database event notification template objects and commanding immediate usage of said server-database template objects by said at least one server, an improved method to maintain operating integrity of said network, said improvement comprising:

said at least one client ensuring that any pre-existing said server database template objects and any of said plurality of template objects which are identically-named contain identical object data.

2. (ORIGINAL) The method of claim 1 and wherein said ensuring comprises:

client retrieving said any pre-existing said server-database template objects and comparing each of the names of said any pre-existing template objects with each of the names of said plurality of template objects stored in said client-database;

adding new templates comprising pre-existing object data associated with any of said names of said any pre-existing template objects that do not match said names of said plurality of template objects, to said client database; and,

resolving any conflict between any said any pre-existing said server-database template objects and any of said plurality of template objects having identical names but having said pre-existing object data different from said object data.

3. (ORIGINAL) The method of claim 2 and wherein said resolving is selected from the group consisting of: deleting said server-stored template objects, renaming said server-stored template objects, and updating said server-stored template objects.

4. (ORIGINAL) The method of claim 2 and further including repeating acts of claim 2 for all said any pre-existing and server-database template objects.

5. (CURRENTLY AMENDED) In a computer network, said network including at least one client having a client-database, at least one server having at least one server-database, said at least one client: (1) selecting [[said]] events and desired notification responses thereto to obtain selected data, (2) creating a plurality of event-notification template objects from said selected data and storing said objects in said client database;

and, (3) transmitting said objects from said client database to selected ones of said at least one server to obtain server-database event notification template objects and commanding immediate usage of said server-database template objects by said at least one server, an improved method for establishing an events notification system, said improvement comprising:

said at least one client ensuring that any pre-existing said server database template objects and any of said plurality of template objects which are identically-named contain identical object data.

6. (ORIGINAL) The method of claim 5 and wherein said ensuring comprises:

client retrieving said any pre-existing said server-database template objects and comparing each of the names of said any pre-existing template objects with each of the names of said plurality of template objects stored in said client-database;

adding new templates comprising pre-existing object data associated with any of said names of said any pre-existing template objects that do not match said names of said plurality of template objects, to said client database; and,

resolving any conflict between any said any pre-existing said server-database template objects and any of said plurality of template objects having identical names but having said pre-existing object data different from said object data.

7. (ORIGINAL) The method of claim 6 and wherein said resolving is selected from the group consisting of: deleting said server-stored template objects, renaming said server-stored template objects, and updating said server-stored template objects.

8. (ORIGINAL) The method of claim 6 and further including repeating acts of claim 6 for all said any pre-existing and server-database template objects.

9. (ORIGINAL) In a client-server computer network, an improved method for handling event-errors occurring at server locations through the reporting of said event-errors to said client and accomplished by (a) creating a client template object containing at least a plurality of event-error types and desired notification procedure for each one of said plurality of event-error types; (b) applying said client template object to each of said server locations to obtain server-location template objects; (c) said server locations instituting said desired notification procedure upon occurrence of said event-errors, said improvement comprising:

purging ambiguities including resolving conflict between said client template object and any pre-existing said server-location template objects.

10. (ORIGINAL) In the method of claim 9, said resolving is selected from the group consisting of : deleting, renaming, and updating said pre-existing server-location template objects.

11. (ORIGINAL) In the method of claim 9, said resolving is selected from the group consisting of deleting, renaming, and updating said client template object.

12. (ORIGINAL) The method of claim 9 and wherein said ambiguity purging comprises:

- (a) said client retrieving said pre-existing said server-location template-objects;
- (b) comparing the names of each of said pre-existing objects with all names in said client template-object;
- (c) adding pre-existing template object data corresponding to any of said names of each of said pre-existing objects that do not match said names in said client template object, to said client template object;
- (d) for any particular said pre-existing objects having the same name as any other contents within said client template object, comparing pre-existing contents of said particular said pre-existing objects with said other contents of said client template-object;
- (e) resolving conflict if said pre-existing contents are different from said other contents; and,
- (f) repeating the acts of retrieving, name-comparing, adding, contents-comparing, and conflict resolving for all said server locations and in said network.

13. (ORIGINAL) The method of claim 12 and wherein said ambiguity purging is selected from the group consisting of: deleting, renaming, and updating said pre-existing server location template objects.

14. (ORIGINAL) The method of claim 12 and wherein said ambiguity purging is selected from the group consisting of: deleting, renaming and updating said client template object.

15. (ORIGINAL) In an improved events notification system deployed across multiple client-server networks including a first client operatively coupled with a first plurality of servers and a second client operatively coupled with a second plurality of servers, said events-notification system employing templates for purposes of (a) communicating operating states to its respective client from its respective servers of peripherals controlled by said its respective servers, (b) selecting type of notification to be initiated under specified failure mode conditions detected in any of said peripherals, and (c) taking action in accordance with said type of notification upon detection of said conditions, said improvement comprising:

under conditions of said first client being operatively coupled to at least one of said second plurality of servers having at least one of said templates of said second plurality of servers in conflict with other of said templates associated with said first client, said first client updating said at least one of said templates of said second plurality of servers to provide an updated template that conforms to said other of said templates.

16. (ORIGINAL) In the improved events-notification system of claim 15 said second client updating certain of said templates in its database which conflict with said updated template to conform to said updated template.

17. (ORIGINAL) An events-notification system utilizing software object templates offering notification options and employable within computer network environments

selected from the group consisting of client-server, SAN, and NAS environments,
comprising:

means for creating said templates and selecting said options to obtain selected
template options at a user interface located at a single location in said network;

means for deploying said templates to network components dispersed throughout
said network from said single location to obtain remotely-deployed templates; and,

means for resolving name/content conflicts between said templates and any other
templates previously deployed and dispersed throughout said network.

18. (ORIGINAL) The system of claim 17 and wherein said creating means includes
means for modifying any of said remotely-deployed templates from said single location.

19. (ORIGINAL) The system of claim 17 and wherein said remotely-deployed
templates, responsive to detection of certain events occurring at said network components
and corresponding to certain of said selected options, notifies said user interface of the
occurrence of said certain events in accordance with notification schemes corresponding
to other of said selected options.

20. (ORIGINAL) The system of claim 19 and wherein said certain events are failure
modes within said network components.

21. (ORIGINAL) The system of claim 17 and wherein said network components are
server groups including processors and storage components.

22. (ORIGINAL) The system of claim 21 and wherein said storage components includes disk arrays.

23. (ORIGINAL) The system of claim 19 and wherein said notification schemes include email transmission, telephone-calling, and telephone-paging.

24. (ORIGINAL) The system of claim 17 and wherein said resolving means includes any of: deleting, renaming and updating said other templates.

25. (ORIGINAL) In a computer network having an events notification system, said network including at least one client having a client-database, at least one server having at least one server-database, said at least one client: (1) selecting said events and desired notification responses thereto to obtain selected data, (2) creating a plurality of event-notification template objects from said selected data and storing said objects in said client database; and, (3) transmitting said objects from said client database to selected ones of said at least one server to obtain server-database event notification template objects and commanding immediate usage of said server-database template objects by said at least one server, improved apparatus to maintain operating integrity of said network, said improvement comprising:

means for ensuring that any pre-existing said server database template objects and any of said plurality of template objects which are identically-named contain identical object data.

26. (ORIGINAL) The apparatus of claim 25 and wherein said means for ensuring comprises:

means for retrieving said any pre-existing said server-database template objects and comparing each of the names of said any pre-existing template objects with each of the names of said plurality of template objects stored in said client-database;

means for adding new templates comprising pre-existing object data associated with any of said names of said any pre-existing template objects that do not match said names of said plurality of template objects, to said client database; and,

means for resolving any conflict between any said any pre-existing said server-database template objects and any of said plurality of template objects having identical names but having said pre-existing object data different from said object data.

27. (ORIGINAL) The apparatus of claim 26 and wherein said means for resolving is selected from the group consisting of: deleting said server-stored template objects, renaming said server-stored template objects, and updating said server-stored template objects.

28. (ORIGINAL) The apparatus of claim 26 and further including means for repeating acts of claim 26 for all said any pre-existing and server-database template objects.

29. (ORIGINAL) In a computer network, said network including at least one client having a client-database, at least one server having at least one server-database, said at least one client employing an events notification system for: (1) selecting said events and desired notification responses thereto to obtain selected data, (2) creating a plurality of event-notification template objects from said selected data and storing said objects in said client database; and, (3) transmitting said objects from said client database to selected ones of said at least one server to obtain server-database event notification template objects and commanding immediate usage of said server-database template objects by said at least one server, an improved computer program product for use with said events notification system, said improvement comprising:

programmable code for ensuring that any pre-existing said server database template objects and any of said plurality of template objects which are identically-named contain identical object data.

30. (ORIGINAL) The computer program product of claim 29 and wherein said programmable code for ensuring comprises:

programmable code for retrieving said any pre-existing said server-database template objects and comparing each of the names of said any pre-existing template

objects with each of the names of said plurality of template objects stored in said client-database;

programmable code for adding to said client database new templates comprising pre-existing object data associated with any of said names of said any pre-existing template objects that do not match said names of said plurality of template objects; and,

programmable code for resolving any conflict between any said any pre-existing said server-database template objects and any of said plurality of template objects having identical names but having said pre-existing object data different from said object data.

31. (ORIGINAL) The computer program product of claim 30 and wherein said programmable code for resolving is selected from the group consisting of: programmable code for deleting said server-stored template objects, programmable code for renaming said server-stored template objects, and programmable code for updating said server-stored template objects.

32. (ORIGINAL) The computer program product of claim 30 and further including programmable code for repeating acts of claim 30 for all said any pre-existing and server-database template objects.

33. (ORIGINAL) In a computer network such as a client-server network, an improved system for handling event-errors occurring at server locations through the reporting of said event-errors to said client and accomplished by (a) creating a client

template object containing at least a plurality of event-error types and desired notification procedure for each one of said plurality of event-error types; (b) applying said client template object to each of said server locations to obtain server-location template objects; (c) said server locations instituting said desired notification procedure upon occurrence of said event-errors, said improvement comprising:

a sub-system for purging ambiguities including resolving conflict between said client template object and any pre-existing said server-location template objects.

34. (ORIGINAL) In the system of claim 33, said sub system for resolving selects from the group consisting of : deleting, renaming, and updating said pre-existing server-location template objects.

35. (ORIGINAL) In the system of claim 33, said sub-system for resolving is selected from the group consisting of deleting, renaming, and updating said client template object.

36. (ORIGINAL) The system of claim 33 and wherein said sub-system for purging ambiguity between said client template object and any pre-existing said server-location template objects comprises:

(a) sub-system components for retrieving said pre-existing said server-location template-objects;

(b) sub-system components for comparing the names of each of said pre-existing objects with all names in said client template-object;

(c) sub-system components for adding pre-existing template object data corresponding to any of said names of each of said pre-existing objects that do not match said names in said client template object, to said client template object;

(d) for any particular said pre-existing objects having the same name as any other contents within said client template object, sub-system components for comparing pre-existing contents of said particular said pre-existing objects with said other contents of said client template-object;

(e) sub-system components for resolving conflict if said pre-existing contents are different from said other contents; and,

(f) sub-system components for repeating the acts of retrieving, name-comparing, adding, contents-comparing, and conflict resolving for all said server locations and in said network.

37. (ORIGINAL) The system of claim 36 and wherein said sub-system components for ambiguity purging select actions from the group consisting of: deleting, renaming, and updating said pre-existing server location template objects.

38. (ORIGINAL) The system of claim 36 and wherein said sub-system components for ambiguity purging is select actions from the group consisting of: deleting, renaming and updating said client template object.

39. (ORIGINAL) In an improved events notification method employable within multiple client-server networks including a first client operatively coupled with a first plurality of servers and a second client operatively coupled with a second plurality of servers, said events-notification method employing templates for purposes of (a) communicating operating states to its respective client from its respective servers of peripherals controlled by said its respective servers, (b) selecting type of notification to be initiated under specified failure mode conditions detected in any of said peripherals, and (c) taking action in accordance with said type of notification upon detection of said conditions, said improvement comprising:

under conditions of said first client being operatively coupled to at least one of said second plurality of servers having at least one of said templates of said second plurality of servers in conflict with other of said templates associated with said first client, said first client updating said at least one of said templates of said second plurality of servers to provide an updated template that conforms to said other of said templates.

40. (ORIGINAL) In the improved events-notification method of claim 39 said second client updating certain of said templates in its database which conflict with said updated template to conform to said updated template.

41. (ORIGINAL) An events-notification computer program product utilizing software object templates offering notification options and employable within computer network

environments selected from the group consisting of client-server, SAN, and NAS environments, comprising:

programmable code for creating said templates and selecting said options to obtain selected template options at a user interface located at a single location in said network;

programmable code for deploying said templates to network components dispersed throughout said network from said single location to obtain remotely-deployed templates; and,

programmable code for resolving name/content conflicts between said templates and any other templates previously deployed and dispersed throughout said network.

42. (ORIGINAL) The computer program product of claim 41 and wherein said programmable code for creating includes programmable code for modifying any of said remotely-deployed templates from said single location.

43. (ORIGINAL) The computer program product of claim 42 including programmable code that is (a) operable with said remotely-deployed templates and (b) responsive to detection of certain events both occurring at said network components and corresponding to certain of said selected options, for notifying said user interface of the occurrence of said certain events in accordance with notification schemes corresponding to other of said selected options.

44. (CURRENTLY AMENDED) The computer program product of ~~claim 43~~claim 43 and wherein said certain events are failure modes within said network components.

45. (ORIGINAL) The computer program product of claim 41 and wherein said network components are server groups including processors and storage components.

46. (ORIGINAL) The computer program product of claim 45 and wherein said storage components includes disk arrays.

47. (ORIGINAL) The computer program product of claim 43 and wherein said notification schemes include email transmission, telephone-calling, and telephone-paging.

48. (ORIGINAL) The computer program product of claim 41 and wherein said programmable code for resolving includes any of: programmable code for deleting, programmable code for renaming and programmable code for updating said other templates.